IN THE CLAIMS

Please cancel Claims 17 – 18 without prejudice, amend Claim 19, and add new Claims 62 – 65 as follows:

1. (Previously presented) A method of analyzing a plurality of input strings of data derived from respective ones of a plurality of data processors, said plurality of input strings of data being generated by a same task, comprising:

initializing the data in each of said strings;

finding the differences between said strings;

providing said differences in a display to a user;

wherein said act of finding the differences comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings; and

analyzing said groups of said data for potential hardware or software integration problems associated with one or more of said data processors based at least in part on said differences between said plurality of strings.

2. (Previously presented) The method of Claim 1, wherein the act of initializing comprises:

creating a symbol table having a plurality of symbol numbers associated therewith;

creating said symbol array having at least one element for each of said strings;

for each of said input strings, determining whether said each string is present in said symbol table; and

if said each string is present in said symbol table, obtaining at least one symbol number for said string from said symbol table.

3. (Previously presented) A method of analyzing a plurality of strings of data derived from at least one data processing device, said plurality of strings of data being generated by a

plurality of processors within said at least one data processing device operating on a same task, comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said strings of data based at least in part on said symbol array; and

identifying at least one relationship between one or more of said data within one or more of said strings;

wherein said act of identifying comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings; and

analyzing said groups of said data for potential hardware or software integration problems associated with one or more of said plurality of processors based at least in part on said relationship between said plurality of strings.

4. (Previously presented) The method of Claim 3, wherein the act of initializing comprises:

creating a symbol table having a plurality of symbol numbers associated therewith;

creating said symbol array having at least one element for each of said strings;

for each of said input strings, determining whether said each string is present in said symbol table; and

if said each string is present in said symbol table, obtaining at least one symbol number for said string from said symbol table.

5. – 10. (Cancelled)

11. (Previously presented) A storage device, comprising:

a computer readable medium;

a computer program stored on said computer readable medium, said program being adapted for analyzing a plurality of strings of data derived from at least one data processing device having a plurality of processor cores, said cores operating on a same task according to the method comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said strings of data based at least in part on said symbol array;

identifying at least one relationship between one or more of said data within one or more of said strings;

wherein said act of identifying comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings; and

analyzing said groups of said data for potential hardware or software integration problems associated with one or more of said plurality of processor cores based at least in part on said relationship between said plurality of strings generated by said same task.

12. (Previously presented) A data processing device, comprising:

a processor adapted to process digital data and execute a computer program;

a storage device in data communication with said processor, said storage device comprising:

a computer readable medium;

a computer program stored on said computer readable medium, said program being adapted for analyzing a plurality of strings of data derived from at least one data processing device having a plurality of processor cores, each of said processor cores running a same task according to the method comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said strings of data based at least in part on said symbol array; and

identifying at least one relationship between one or more of said data within one or more of said strings;

wherein said act of identifying comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings; and

analyzing said groups of said data for potential hardware or software integration problems associated with one or more of said plurality of processor cores based at least in part on said relationship between said plurality of strings generated by said same task.

13. (Previously presented) The device of Claim 12, further comprising a plurality of data interfaces adapted to receive said data strings from respective ones of a plurality of software processes running on respective ones of said plurality of processor cores.

14. – 18. (Cancelled)

19. (Currently amended) A multi-processor integrated circuit device, comprising:

a first processor core adapted to run a first software process, said first software process adapted to generate a first string of data;

a second processor core adapted to run a second software process said second software process adapted to generate a second string of data;

wherein said first and second software processes are a same task;

at least one data interface, wherein said first and second processors respectively transfer data comprising said first and second strings to an external debug process adapted to identify similarities and differences between the operation of said first and second software processes on said first and second processors, via said at least one interface;

wherein said debug process identifies the similarities and differences by:

identifying groups of said data within said first and second strings that are identical across at least both of said strings; and

identifying groups of said data within said first and second strings that appear in the same order within at least both of said strings; and

wherein said debug process is adapted to analyze ~~analyzing~~ said groups of said data for potential hardware or software integration problems associated with either of said processor cores based at least in part on said similarities and differences between said plurality of strings generated by said first and second software processes.

20. (Previously presented) The method of Claim 1, wherein said strings each comprise a plurality of lines, and said method further comprises:

forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively analyzing, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

21. (Previously presented) The storage device of Claim 11, wherein said strings each comprise a plurality of lines, and said method further comprises:

forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively analyzing, in order: a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

22. (Previously presented) The integrated circuit device of Claim 19, wherein said strings each comprise a plurality of lines, and said debug process is further adapted to:

form a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively evaluate, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

23. – 49. (Cancelled)

50. (Previously presented) A method of analyzing a plurality of inputs of data derived from respective ones of a plurality of data processors, comprising:

initializing the data in each of said inputs;

generating said plurality of inputs of data by running a same task on respective ones of said plurality of data processors;

identifying at least one relationship between one or more of said data within one or more of said inputs, said act of identifying comprising:

creating a list of anchors containing strings that occur exactly once in every one of said inputs;

marking at least one of said strings as belonging to a group; and

determining whether a string occurring before said group is identical in each of said inputs; and

analyzing said at least one relationship among said plurality of inputs of data, said act of analyzing comprising finding areas of the design which may require refinement or verification.

51. (Previously presented) The method of Claim 50, further comprising examining a string immediately after said group to determine if it is identical in each of said inputs.

52. – 53. (Cancelled)

54. (Previously presented) A method of analyzing a plurality of input strings of data derived from respective ones of a plurality of data processors, said plurality of input strings of data being generated by a same task, comprising:

varying at least one parameter on at least one of said plurality of data processors;

initialize the data in each of said strings;

find the differences between said strings generated by said same task;

provide said differences in a display to a user;

wherein said act of finding the differences comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings; and

analyzing said groups of said data to verify that said act of varying at least one parameter on at least one of said plurality of data processors remains within a desired parameter.

55. (Previously presented) The method of Claim 54, wherein said plurality of data processors are RISC cores.

56. (Previously presented) The method of Claim 54, wherein at least one of said plurality of data processors is a RISC core and at least one of said plurality of data processors is a digital signal processor.

57. (Previously presented) The method of Claim 54, wherein at least one of said plurality of data processors is a RISC core and at least one of said plurality of data processors is an ASIC macro-function.

58. (Previously presented) An apparatus for simulating the operation of a multi-processor integrated circuit device, comprising:

a first simulated processor core adapted to run a first software process, said first software process adapted to generate a first string of data;

a second simulated processor core adapted to run a second software process said second software process adapted to generate a second string of data;

wherein said first and second software processes are the same;

a debug process adapted to identify similarities and differences between the operation of said first and second software processes on said first and second simulated processors, via at least one interface;

wherein said debug process identifies the similarities and differences by:

identifying groups of said data within said first and second strings that are identical across at least both of said strings; and

identifying groups of said data within said first and second strings that appear in the same order within at least both of said strings.

59. (Previously presented) The apparatus of Claim 58, wherein said strings each comprise a plurality of lines, and said simulation further comprises:

forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively analyzing, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

60. (Previously presented) A method for migrating a program from a first processor to a second processor, comprising:

disposing said program on said first processor;

disposing said program on said second processor;

debugging said program on both said first and second processors simultaneously, said act of debugging, comprising:

generating a plurality of strings of data derived from respective ones of said processors;

finding the differences between said strings; and

providing said differences in a display to a user;

wherein said act of finding the differences comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings; and

analyzing said groups of said data for potential hardware or software integration problems associated with said program migration from said first processor to said second processor based at least in part on said differences between said plurality of strings.

61. (Previously presented) A method of analyzing a plurality of input strings of data derived from respective ones of a plurality of data processors, said plurality of input strings of data being generated by a same task, comprising:

varying at least one parameter on at least one of said plurality of data processors;

initialize the data in each of said strings;

find the differences between said strings; and

provide said differences in a display to a user;

wherein said act of finding the differences comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings; and

analyzing said groups of said data to verify that said act of varying at least one parameter on at least one of said plurality of data indicates some improvement in said act of generating.

62. (New) An integrated circuit device, comprising:

a first processing means for running a first software process, said first software process adapted to generate a first string of data;

a second processing means adapted to run a second software process said second software process adapted to generate a second string of data;

wherein said first and second software processes are a same task;

at least one data interface means, wherein said first and second processing means respectively transfer data comprising said first and second strings to an external debug process adapted to identify similarities and differences between the operation of said first and second software processes on said first and second processing means, via said at least one interface means;

wherein said debug process identifies the similarities and differences by:

identifying groups of said data within said first and second strings that are identical across at least both of said strings; and

identifying groups of said data within said first and second strings that appear in the same order within at least both of said strings; and

wherein said debug process is adapted to analyze said groups of said data for potential hardware or software integration problems associated with either of said processor cores based at least in part on said similarities and differences between said plurality of strings generated by said first and second software processes.

63. (New) The integrated circuit device of Claim 62, wherein said strings each comprise a plurality of lines, and said debug process is further adapted to:

form a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively evaluate, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

64. (New) An apparatus for simulating the operation of an integrated circuit device, comprising:

a first simulated processing means for running a first software process, said first software process adapted to generate a first string of data;

a second simulated processing means for running a second software process said second software process adapted to generate a second string of data;

wherein said first and second software processes are the same;

a debug processing means for identifying similarities and differences between the operation of said first and second software processes on said first and second simulated processing means, via at least one interface;

wherein said debug processing means identifies the similarities and differences by:

identifying groups of said data within said first and second strings that are identical across at least both of said strings; and

wherein said debug processing means is adapted to identify groups of said data within said first and second strings that appear in the same order within at least both of said strings.

65. (New) The apparatus of Claim 64, wherein said strings each comprise a plurality of lines, and said simulation further comprises:

forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively analyzing, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.